

## 2.1.1 Teste cu alegere multiplă și duală

1. Ce se va afișa în urma executării programului următor:

```
var x,y,z:byte;
procedure P(a,b:byte;var c:byte);
var aux:byte;
begin
  aux:=a; a:=b-a;
  c:=c-aux; b:=aux;
end;
begin
  x:=10; y:=100; z:=200;
  p(x,y,z);
  write(x, ' ', y, ' ', z, ' ');
  p(x,y,z);
  write(x, ' ', y, ' ', z);
end.
```

```
#include <iostream.h>
int x,y,z;
void P(int a,int b, int &c)
{ int aux;
  aux=a; a=b-a ;
  c=c-aux ; b=aux;
}

void main ()
{ x=10; y=100; z=200;
  P(x,y,z);
  cout<<x<<" " <<y<<" " <<z<<" ";
  P(x,y,z);
  cout<<x<<" " <<y<<" " <<z<<" ";
}
```

- a) 90 10 190 176 90 100
- b) 10 10 190 10 10 180

- c) 10 100 190 10 100 190
- d) 10 100 190 10 100 180

2. Ce se va afișa în urma executării programului următor:

```
var a:byte;
procedure P(x:byte; var y:byte);
begin
  y:=y*x; x:=x+y;
  write(x, ' ', y, ' ')
end;
begin
  a:=11;
  p(a,a);
  writeln(a);
end.
```

```
#include <iostream.h>
int a =11;
void P(int x,int &y)
{
  y=y*x; x=x+y;
  cout<<x<<" " <<y<<" ";
}

void main(){
  P(a,a);
  cout<<a;
}
```

- a) 11 121 11
- b) 132 121 121

- c) 242 242 242
- d) 11 121 121

3. În urma apelului  $P(n,x)$  se dorește afișarea cifrei maxime a unui număr natural prin intermediul apelului  $write(x)$ , respectiv  $cout<<x$ . Identificați antetul corect al subprogramului  $P$ .

a)  
**function**  $P(n:integer):byte$ ;

b)  
**procedure**  $P(var a,b:integer)$ ;

c)  
**procedure**  $P(n,x:integer)$ ;

d)  
**procedure**  $P(x:integer;var n:byte)$ ;

a)  
**int**  $P(int n)$

b)  
**void**  $P(int \&x,int \&y)$

c)  
**void**  $P(int n, int x)$

d)  
**void**  $P(int x, int \&n)$

4. Subprogramul  $P$  efectuează ștergerea dintr-un vector, transmis ca parametru, a tuturor elementelor egale cu o valoare cunoscută. Subprogramul primește, prin intermediul altor doi parametri întregi, lungimea tabloului și valoarea elementelor ce vor fi șterse. Identificați antetul corect al subprogramului  $P$ . Pentru varianta Pascal, se consideră următoarea definiție de tip: *type sir=array[1..100] of integer*.

a)  
**function**  $P(a:sir;$   
                   $n,x:integer):sir$ ;

b)  
**procedure**  $P(var a:sir;$   
                   $n,x: integer)$ ;

c)  
**procedure**  $P(var a:sir;var n:byte;$   
                   $x:integer)$ ;

d)  
**procedure**  $P(a:sir; n,x:integer)$ ;

a)  
**int**  $P(int a[100], int n, x)$

b)  
**void**  $P(int a, int \&x, int \&y)$

c)  
**void**  $P(int a[100],int \&n, int x)$

d)  
**void**  $P(int a[100], int x, int n)$

5. Ce valori vor fi afișate în urma executării următorului program?

```
var x:integer;

procedure p1(y:integer);
var x:integer;
begin
x:=y; x:=2; write(x, ' '); y:=x;
end;

procedure p2(var y:integer);
begin
x:=y; x:=3; write(x, ' '); y:=x;
end;
```

```
#include <iostream.h>
int x;
void p1(int y)
{ int x;
  x=y; x=2;
  cout<<x<<' '; y=x ;
}

void p2(int &y)
{ x=y; x=3;
  cout<<x<<' '; y=x;
}
```

```

begin
  x:=1;
  p1(x); write(x, ' ');
  p2(x); writeln(x);
end.

```

- a) 2 2 3 3  
b) 2 2 3 2

```

void main()
{ x=1;
  p1(x); cout<<x<<' ';
  p2(x); cout<<x<<endl;
}

```

- c) 2 1 3 3  
d) 2 1 3 2

6. Ce valori vor fi afișate în urma executării următorului program?

```

var x:integer;

procedure p1(x:integer);
var y:integer;
begin
  y:=10; x:=x + y; write(x, ' ');
end;

procedure p2(var x:integer);
var y:integer;
begin
  y:=x; x:=x + 3;
  x:=x + y; write(x, ' ');
end;

begin
  x:=20;
  p1(20); write(x, ' ');
  p2(x); write(x, ' ');
end.

```

- a) 20 20 43 20  
b) 30 20 43 43

```

#include <iostream.h>
int x;
void p1(int x)
{ int y=10;
  x=x+y;
  cout<<x<<' ';
}

void p2(int &x)
{int y;
  y=x; x+=3; x+=y;
  cout<<x<<' ';
}

void main()
{ x=20;
  p1(20); cout<<x<<' ';
  p2(x); cout<<x<<endl;
}

```

- c) 30 20 43 20  
d) 20 20 46 46

7. Ce valori vor fi afișate în urma executării următorului program?

```

var x,y:integer;

procedure p(var a,b:integer);
var x:integer;
begin
  x:=a*b; a:=a+x; b:=a+x;
  write(a, ' ', b, ' ', x, ' ');
end;

begin
  x:=5; y:=10;
  p(x,y); write(x, ' ', y);
end.

```

- a) 55 105 50 105 105  
b) 55 55 50 55 55

```

#include <iostream.h>
int x,y;

void p(int &a,int &b)
{ int x;
  x=a*b; a+=x; b+=a+x;
  cout<<a<<' '<<b<<' '<<x<<' ';
}

void main()
{ x=5; y=10;
  p(x,y); cout<<x<<' '<<y;
}

```

- c) 55 105 50 55 55  
d) 55 105 50 55 105

8. Ce valori vor fi afișate în urma executării următorului program?

```
var n:integer;
function cif(var x:integer):byte;
var c:byte;
begin
  c:=x mod 10;  x:=x div 10;
  cif:=(c+x mod 10)mod 10;
end;
begin
  n:=21987;
  writeln(cif(n)+cif(n))
end.
```

```
#include <iostream.h>
int n=21987;
int cif(int &x)
{
  int c;
  c=x%10; x/=10;
  return (c+ x%10)%10;
}
void main()
{ cout << cif(n)+cif(n);
}
```

- a) 12;            b) 10;                            c) 32;                            d) 30.

9. Indicați care dintre următoarele antete de funcții sunt corecte sintactic:

- a) `function F1(z byte);`  
 b) `function F2(x,y):real;`  
 c) `function F3(y:byte):text;`  
 d) `function 12F(c:char):real;`  
 e) `function F4(y:byte;z:byte):real;`

- a) `int F1(int z;)`  
 b) `int F2(int x,y)`  
 c) `FILE* F3(int y)`  
 d) `float 12F(char c)`  
 e) `float F4(int x, int y)`

10. Identificați care dintre funcțiile următoare returnează un număr real, reprezentând media aritmetică dintre câtul și restul la împărțirea a două numere întregi, transmise ca parametri:

a) `function M(x,y:integer):byte;`  
`begin`  
`M := (x div y + x mod y)/2;`  
`end;`

b) `function M(x,y:integer):real;`  
`var medie:real;`  
`begin`  
`medie := x div y + x mod y;`  
`M := medie/2;`  
`end;`

c) `function M(x,y:integer):real;`  
`begin`  
`M := x div y + x mod y;`  
`M := M/2;`  
`end;`

a) `int M(int x,y)`  
`{`  
`return (x/y + x%y)/2;`  
`}`

b) `float M(int x,int y)`  
`{`  
`float medie;`  
`medie = x/y + x%y;`  
`return medie/2;`  
`}`

c) `float M(int x,int y)`  
`{`  
`return x/y + x%y;`  
`}`

```

d)
function M(x,y:integer):real;
begin
  x := x div y;
  y := x mod y;
  M := (x + y)/2;
end;

```

```

d)
float M(int x,int y)
{
  x = x/y ;
  y = x*y ;
  return (x+y)/2;
}

```

11. Considerăm definită funcția *prim*, care returnează *true* în varianta Pascal, respectiv 1 în varianta C++, dacă valoarea întreagă transmisă ca parametru este un număr prim și *false* / 0, în caz contrar. Considerăm un tablou unidimensional *a* ce conține *n* elemente de valori întregi. Identificați care dintre următoarele funcții returnează indicele din *a* al primului element număr prim. Dacă acesta nu conține nici un element prim, se returnează valoarea -1.

Pentru varianta Pascal, se consideră următoarea definiție de tip: *type sir=array[0..100] of integer*, deci primul element este *a[0]*.

```

a)
function p(a:sir;n:byte):integer;
var i:integer;
begin
  p:=-1;
  for i:=0 to n-1 do
    if prim(a[i]) then p:=i;
  end;

```

```

b)
function p(a:sir;n:byte):integer;
var x:integer;
begin
  x:=-1;
  repeat
    inc(x)
  until (prim(a[x])) or (x=n);
  if x=n then p:=-1
    else p:=x;
end;

```

```

c)
function p(a:sir;n:byte):integer;
var x,i:integer;
begin
  x:=-1;
  while x<n-1 do begin
    inc(x);
    if prim(a[x])=true then x:=n;
  end;
  p:=x;
end;

```

```

a)
int p1(int a[101], int n)
{
  int i,x=-1;
  for(i=0;i<n;i++)
    if (prim(a[i])) x=i;
  return x;
}

```

```

b)
int p2(int a[101], int n)
{
  int x=-1;
  do x++;
  while (!prim(a[x]) && (x<n));
  if (x==n) return -1;
  else return x;
}

```

```

c)
int p3(int a[101], int n)
{
  int i,x=-1;
  while (x<n-1) {
    x++;
    if (prim(a[x])) return x;
  }
  return x-n-1;
}

```

```

begin
  a:='copiii'; b:='i';
  x(a,b);
  writeln(a,b);
end.

```

```

void main() {
  a = "copiii"; b = 'i';
  x(a,b);
  cout<<a<<b<<endl;
}

```

- a) copP                      b) copiii.                      c) copiiii                      d) copiiiP

15. Care dintre următoarele funcții returnează cea mai mare putere a lui 2 care este mai mică sau egală cu o valoare naturală (<15), transmisă ca parametru?

a)

```

function F(n:byte):integer;
var x:integer;
begin
  x:=-1;
  repeat
    inc(x);
  until 1 shl x>n;
  F:=1 shl (x-1);
end;

```

b)

```

function F(n:byte):integer;
var x:integer;
begin
  x:=1;
  while (x<n) do x:=x*2;
  F:=x;
end;

```

c)

```

function F(n:byte):integer;
var x:integer;
begin
  x:=1;
  repeat
    x:=x*2;
  until x>=n;
  F:=x-1;
end;

```

d)

```

function F(n:byte):integer;
var i,x:integer;
begin
  x:=1;
  for i:=1 to n do begin
    x:=x*2;
    if x<=n then begin
      F:=x; exit;
    end;end;
  F:=x-1;
end;

```

a)

```

int F(int n)
{
  int x=-1;
  do
    x++;
  while (1 << x <= n );
  return 1 << (x-1);
}

```

b)

```

int F(int n)
{
  int x=1;
  while (x<n) x*=2;
  return x;
}

```

c)

```

int F(int n)
{
  int x=1;
  do
    x*=2;
  while (x<n);
  return x-1;
}

```

d)

```

int F(int n)
{
  int i,x;
  x=1;
  for(i=1;i<=n;i++)
  {
    x*=2;
    if (x<=n) return x;
  }
  return x-1;
}

```

16. Ce se va afișa în urma executării următorului program:

```
var a:real;
function F(var x:real):real;
begin
    x:=abs(x*10);
    F:=int(x/10) + frac(x);
end;
begin
    a:=15.25;
    write(F(a):0:2, ' ');
    write(F(a):0:2, ' ');
    writeln(a:0:2);
end.
```

```
#include <stdio.h>
float F(float &x)
{ x=(x > 0 ? x: -(x));
  x*=10;
  return int(x/10) + x - int(x);
}
void main()
{float a=15.25;
 printf("%0.2f ",F(a));
 printf("%0.2f ",F(a));
 printf("%0.2f ",a);
}
```

- a) 15.50 15.50 1525.00  
b) 15.50 15.50 152.50

- c) 15.50 152.00 1525.00  
d) 15.50 152.00 152.50

17. Câte elemente divizibile cu 10 se vor afișa în urma executării programului următor:

```
var x,y:byte;
function F(var y:byte;x:byte):byte;
begin
    y:=y div 10 + x;
    F:= x+y ;
end;
begin
    x:=101; y:=10;
    write(F(x,y), ' ');
    write(x, ' ', y, ' ');
    writeln(F(x,y));
end.
```

```
#include <iostream.h>
int F(int &y, int x)
{
    y = y/10 + x;
    return x + y;
}
main()
{int x = 101, y = 10;
 cout<<F(x,y)<<" ";
 cout << x <<" "<< y<<" ";
 cout<<F(x,y);
}
```

a) 1

b) 2

c) 3

d) 4

18. În urma apelului  $write(P(n, x))$ , respectiv  $cout<<P(n, x)$ , se dorește afișarea numărului de apariții al cifrei  $x$  în scrierea zecimală a numărului întreg  $n$ . Identificați antetul corect al subprogramului  $P$ .

a)  
**procedure**  $P(n, x: integer)$

b)  
**function**  $P(x, c: integer): integer$

c)  
**function**  $P(n: real; x: byte): byte$

d)  
**function**  $P(x: integer; n: char): byte$

a)  
**void**  $P(int\ n, int\ x)$

b)  
**int**  $P(int\ x, int\ c)$

c)  
**int**  $P(int\ n[10], int\ x)$

d)  
**void**  $P(int\ x, float\ n)$

19. Considerăm subprogramul următor:

<pre> <b>function</b> P(<b>var</b> x:integer):byte; <b>var</b> s:byte; <b>begin</b>     s:=0;     <b>while</b> x&gt;0 <b>do begin</b>         s:=s+x mod 2;  x:=x div 2;     <b>end;</b>     P:=s; <b>end;</b> </pre>	<pre> <b>int</b> P(<b>int</b> &amp;x) { <b>int</b> s=0;   <b>while</b> (x!=0)   {     s+=x%2;     x/=2;   }   <b>return</b> s; } </pre>
---	---

Considerăm că variabila întreagă  $n$ , declarată global, are valoarea 32. Ce se va afișa în urma apelului  $\text{write}(P(n)+P(n)+1)$ , respectiv  $\text{cout}<<P(n)+P(n)+1$ ?

- a) 2;                      b) 3;                      c) 4;                      d) 1.

20. Considerăm secvența de declarații următoare:

<pre> <b>var</b> x:array[1..10]<b>of</b> byte;  <b>procedure</b> P1(n:integer); <b>var</b> x:integer; <b>begin</b>     ... <b>end;</b>  <b>procedure</b> P2(n,x:real); <b>begin</b>     ... <b>end;</b>  <b>begin</b>     ... <b>end.</b> </pre>	<pre> <b>int</b> x[100];  <b>void</b> P1(<b>int</b> n) { <b>char</b> x;   ... }  <b>void</b> P2(<b>float</b> n,<b>float</b> x) {   ... }  <b>void</b> main () {   ... } </pre>
--	--

Care dintre următoarele afirmații este adevărată:

- a) Elementul  $x[1]$  poate fi referit oriunde în program, deci în oricare subprogram  $P1$  sau  $P2$ ;  
b) Programul sursă conține erori de sintaxă;  
c) Elementul  $x[1]$  poate fi referit doar în programul principal (varianta Pascal), respectiv numai în funcția  $\text{main}$  (varianta C++);  
d) Elementul  $x[1]$  poate fi referit doar în subprogramele  $P1$  și  $P2$ .